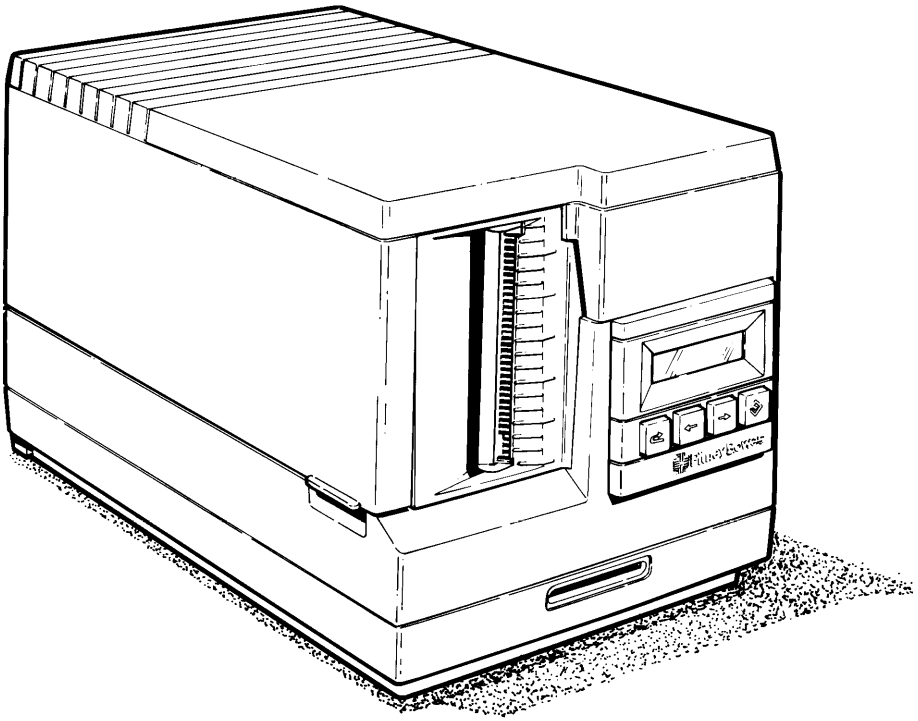
Monarch
Marking Systems

Monarch® 9406™ Thermal Printer

Graphics Manual



TC9406GR Rev. B 8/94
©1994 Monarch Marking Systems, Inc. All rights reserved.

Each product and program carries a respective written warranty, the only warranty on which the customer can rely. Monarch reserves the right to make changes in the product and the programs and their availability at any time and without notice. Although Monarch has made every effort to provide complete and accurate information in this manual, Monarch shall not be liable for any omissions or inaccuracies. Any update will be incorporated in a later edition of this manual.

WARNING

This equipment can interfere with radio communication if not installed and used in accordance with the instruction manual. It has been tested and found to comply with the limits for a Class A computing device pursuant to FCC Rules, Subpart B, Part 15, which provide reasonable protection against such interference when operated in a commercial environment. Operation in a residential area is likely to cause interference which the user must correct at their own expense.

CANADIAN D.O.C. WARNING

This digital apparatus does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe A prescrites dans le Réglement sur le brouillage radioélectrique édicte par le ministère des Communications du Canada.

Trademarks

Monarch is a registered trademark and 9406 is a trademark of Monarch Marking Systems, Inc.
MS-DOS is a registered trademark of Microsoft Corporation.

Table of Contents

1. Introduction to MPCL Graphics 1-1
Who Should Read This Manual 1-1
How Images are Printed 1-2
Choosing a Method 1-3
Hexadecimal Method 1-3
Run Length Encoding Method 1-3

2. Creating the Graphic Image 2-1
Mapping Graphic Images 2-1
Hexadecimal Method 2-1
Run Length Encoding 2-4
Storing Graphic Images 2-6
Using RAM 2-6
Using Temporary Storage 2-7

3. Defining Graphic Images 3-1
Graphic Header 3-1
Bitmap Field Types 3-4
Bitmap Fields 3-5
Next-Bitmap Field 3-6
Duplicate Field 3-8
Constant Text Fields 3-9
Line Fields 3-11
Box (Quadrilateral) Field 3-13
Example Bitmap 3-15

4. Placing the Graphic in a Format 4-1

Appendix A. Conversion Charts A-1
Binary to Hex Conversion Chart A-1
Run Length Encoding Chart A-6

1. Introduction to MPCL Graphics

You can create graphic images for printing on Monarch® 9406™ printers.

Monarch Printer Control Language (MPCL) graphics provide a means of printing logos and other graphic images in label or tag formats.

There are two steps to designing a graphic for printing on a Monarch printer.

1. Create a file from the graphic image.

2. Place the image in a format.

Chapters 2 and 3 tell how to construct an image.

Chapter 4 tells how to place the image in a format.

Who Should Read This Manual

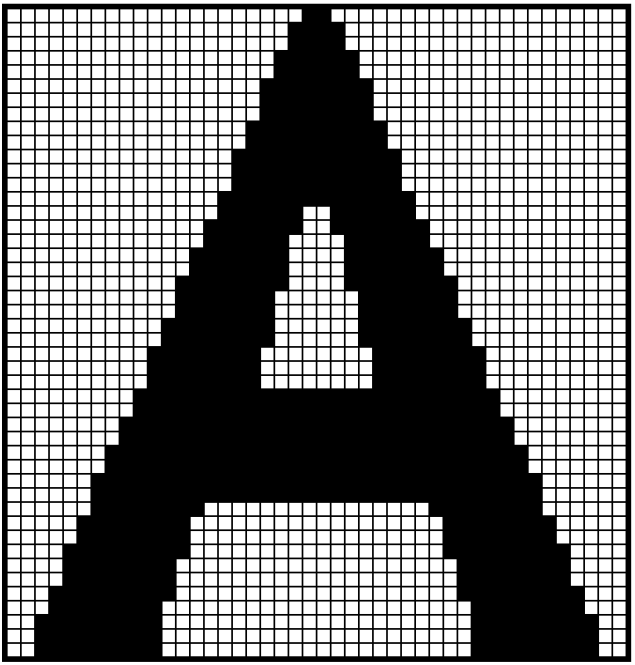
This manual is written for programmers and experienced users of MS-DOS computer systems.

The information presented in this manual deals with binary and hexadecimal numbers. You should be familiar with these number systems to create files for graphics.

How Images are Printed

Printed images on Monarch printers are formed by a system of dots, as shown below.

NOTE: You need to create the graphic image upside down. It will print right side up.



Each square on the grid represents a dot on the printhead. You create a graphic image by blackening dots in a specific pattern. The example above shows how black and white dots form the letter A.

This process is referred to as turning dots "ON" or "OFF." A black dot is "ON." A sequence of binary numbers, called a bit pattern or bitmap, determines what dots are on and off. The numbers "0" and "1" are used for this purpose. The number "1" turns a dot on and "0" turns a dot off.

NOTE: Do not exceed 50% coverage per square inch.

Choosing a Method

You can use two methods to map your graphic image. The most efficient encoding method depends on how complicated your graphic image is and whether or not imaging time is a concern. You may want to experiment with both encoding methods to get optimal performance.

Hexadecimal Method

With the Hexadecimal method, the dot sequences are segmented into binary numbers, then converted to hexadecimal numbers.

A graphic using gray-scaling, several slanted lines, or several vertical lines typically translates more efficiently with the Hexadecimal method.

Run Length Encoding Method

With the Run Length Encoding method, the dots are segmented within a row into sequences where all of the dots are on or all of the dots are off. Then, you convert the strings into a letter. This method is more complicated, but can reduce imaging time for graphics that contain repetitive rows of dots.

A graphic with horizontal lines or very few white-to-black borders typically translates more efficiently with the Run Length Encoding method.

2. Creating the Graphic Image

This chapter describes how to

- map a graphic image.
- store a graphic image.

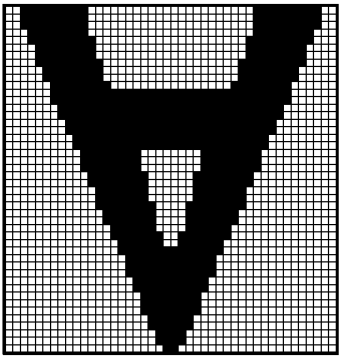
Mapping Graphic Images

To map a graphic image, you can use either the Hexadecimal or Run Length Encoding method.

Hexadecimal Method

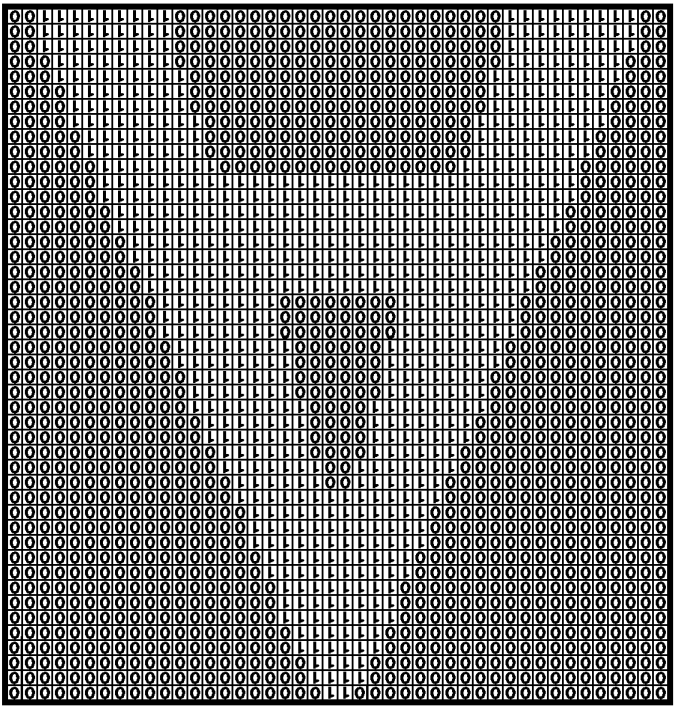
The following sequence explains how to derive a hexadecimal character string from a bitmap image.

1. Refer to the image on the grid.



2. Assign the number 1 to every black square and 0 to every white square.

NOTE: Although you must create these images upside down, they will print right side up.



3. Write the first row of numbers in sequence, as they correspond to the squares in the grid.

```
000000000000000000000011000000000000000000000000
```

4. Divide the row of numbers into groups of eight. If the last group has less than eight numbers, add enough 0's to make a group of eight.

```
00000000 / 00000000 / 00000110 / 00000000 /
              000000 / 0000<u>0000</u>
```

5. Convert each group of eight binary digits to hexadecimal values. Use a calculator for binary to hex conversion, or consult the binary to hex conversion chart in Appendix A.

```
00000000 = 00
00000000 = 00
00000110 = 06
00000000 = 00
00000000 = 00
00000000 = 00
```

6. Write the hexadecimal values in sequence, uninterrupted, for each row. All hex numbers must have two digits. For example, write hex 0 as 00, or hex E as 0E.

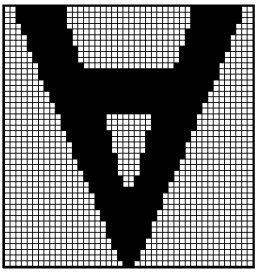
```
00 00 06 00 00 00 =
    000006000000
```

7. Repeat steps **4** through **6** for each row on the grid.

8. Insert the hexadecimal values in the bitmap field definition as shown in Chapter 3, "Defining Graphic Images."

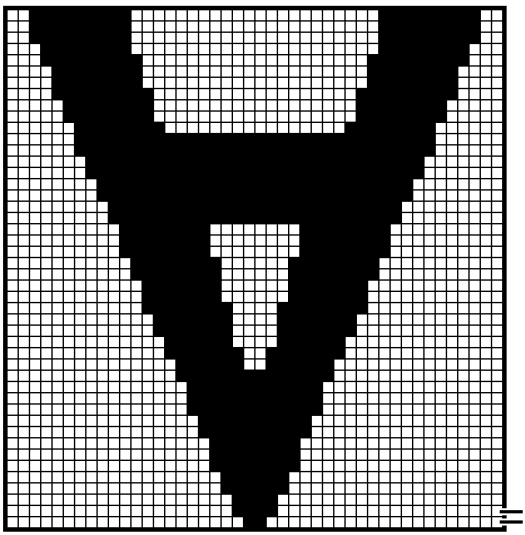
Run Length Encoding

1. Refer to the image on the grid.



2. In the first row, count the number of consecutive dots of the same color. Write the number of consecutive white dots, then the number of consecutive black dots, and so on, in sequence for the first row on the grid. Write "White" or "Black" after each number to indicate white or black dots.

NOTE: Although you must create these images upside down, they will print right side up.



20 White, 4 Black, 20 White
21 White, 2 Black, 21 White

3. Using the Run Length Encoding chart in Appendix A, replace each number you have written with its corresponding code. Be sure to use *CAPITAL* letters for *black* dots and *lower-case* letters for *white* dots.

If the number is greater than 26, write **Z**, followed by the letter corresponding to the amount over 26. For example, to represent 30 dots, write **ZD**.

```
1.  21 white = u, 2 black = B, 21 white = u
2.  20 white = t, 4 black = D, 20 white = t
                    .
                    .
                    .
```

4. Write the letter codes in sequence, uninterrupted, for each row.

```
1.  uBu
2.  tDt
        .
        .
        .
```

NOTE: If the end of the line specifies OFF dots (lower-case letters), the ending lower-case letters can be omitted.For example, uBu can be written as uB.

5. Repeat steps **2** through **4** for each row on the grid.

6. Insert the code values in the bitmap field definition as shown in Chapter 3, "Defining Graphic Images."

Storing Graphic Images

When you have mapped your graphic image, determine how you want to store it. You have two options:

- RAM
- Temporary Storage

Using RAM

You can store small graphics less than approximately 1/2 inch by 1/2 inch in printer RAM and reference them by the graphic ID number.

The printer has 57K of RAM available for storing downloaded data. Downloaded data can be in the form of formats, batches, check digits, or graphics. To determine if you have enough room to store the graphic image in RAM:

1. Measure the image's length and width in inches.

2. Use the following formula to determine the size, in kilobytes, of your graphic image:

```
(9 * W + 2.7) * L
```

where *L* is the length and *W* is the width in inches.

For example, if it is .25 inches wide and .4 inches tall, it would use 1.98K of RAM.

If the size of the graphic image is greater than the amount of available memory in RAM, you must send the graphic image to temporary storage. Use the **T** option in parameter 4 (device) in the Graphics Header.

Using Temporary Storage

You can store graphics up to 1536 rows long with up to 768 dots in each row and refer to them by the graphic ID number.

The printer holds graphic data in temporary storage until you download a new temporary graphic or you use a different format. You can continue to download batch information without risk to the graphic as long as the batches use the current format.

NOTE: If you store a graphic in temporary storage, do not place a graphic field definition in the format (as shown in Chapter 4, "Placing the Graphic in a Format"). This will cause an error.

To determine if you have enough room to store the graphic image in temporary storage:

1. Measure the image's length and width in inches.

2. Use the following formula to determine the size, in kilobytes, of your graphic image:

```
L * W * 4.5
```

where *L* is the length and *W* is the width in inches.

For example, if the graphic is .75 inches tall and .6 inches wide, it would use 2.025K of temporary storage.

3. Defining Graphic Images

Your graphic file can contain

- bitmaps (for detailed graphics)
- constant text
- lines
- boxes.

Images using Hexadecimal Representation or Run Length Encoding are bitmap images. Refer to Chapter 2, "Creating the Graphic Image" to map your graphic image.

When you have mapped your graphic image, you are ready to define a graphic file. This file generates the graphic image that you use in a format.

Graphic Header

Every graphic file must contain a Graphic Header. This record is the first line you enter in a graphic file. It identifies the graphic and provides important measurement and formatting information.

Bitmap, constant text, line, and box fields follow the Graphic Header, if you use them.

Syntax

```
{G,graphID,action,device,units,row,col,mode,name ¦
```

{ Left brace is the starting token for MPCL packets.

G Token to begin a Graphics Header.

graphID A number from 1 to 30 to identify the graphic image.

action Function to perform on the graphic.

- **A** Add to printer memory
- **C** Clear from printer memory

device Printer device that will store the graphic.

- **R** Printer RAM. Stored permanently in the printer.
- **T** Temporary storage. Stored until you reference another graphic.

NOTE: Graphics require large amounts of RAM. When using graphics larger than 1/2 inch by 1/2 inch, Temporary storage is recommended.

units Unit of measure for the graphic. Units selected need not match the format in which the graphic appears. All units used in the graphic must use the units selected here.

- **G** Dots. This value is required for any graphic using bitmap fields. Dot measurement is recommended for all graphics. It provides the most precise graphic measurement. The 9406 prints 192 dots per inch.
- **E** English. Measured in 1/100 inches.
- **M** Metric. Measured in 1/10 millimeters.

row Distance between the *bottom* of the print area and the bottom of the graphic image. Must be measured in units specified above.

col Distance between the *left edge* of the print area and the left edge of the graphic image. Must be measured in units specified above.

NOTE: The first dot on the printhead starts 0.060" from the left edge of the supply.

mode Imaging mode. Enter 0 in this parameter. 0 is the only value supported on the 9406 printer.

name Assign a name to the graphic image. The name must be 1 to 8 characters in length. Enclose in quotation marks. Example: "Name"

Example

```
{G,5,A,R,G,96,48,0,"Letter A"
```

- **G**raphics Header
- Image is identified by number **5**
- Image being **A**dded
- Image stored in **R**AM
- Image uses **G**raphic (dot) measurement
- Image will be placed **96** dots (1/2 inch) from bottom of print area
- Image will be placed **48** dots (1/4 inch) from left edge of print area
- Imaging mode is **0**
- Image is called **Letter A**.

NOTE: The printer uses the row and column locations in the Graphics Header only if a format does not reference this graphic.

If a format references this graphic image (see Chapter 4, "Placing the Graphic in a Format), the graphic field in the format determines the row and column locations.

Bitmap Field Types

When you map the image, use bitmap methods to create detailed graphic images, such as logos.

There are three types of bitmap fields, and each represents one row of dots on the image.

Bitmap	Bitmap field establishes the location and type of coding used to define the image. Typically, you use this type of field for the first field only. Use the simpler "Next-Bitmap" syntax for later fields in the graphic.
Next-Bitmap	This field uses the bitmap information like the bitmap field, but you place it with respect to the preceding bitmap or next-bitmap field. Doing so eliminates the need to re-calculate the location with respect to the supply.
Duplicate	If a bitmap field is identical to a previous bitmap field, the duplicate field allows you to duplicate the field without retyping the data.

Bitmap Fields

Use this record for only the first field. Use the simpler Next-Bitmap field for later fields in the graphic.

Syntax

```
B,row,column,algorithm,"data" ¦
```

B — Token to begin a bitmap field.

row — Distance from the reference *row* selected in the Graphics Header to the *bottom* of the graphic image. Measure in units selected in the Graphics Header: 1/100 inches (English), 1/10 millimeters (Metric), or dots (Graphic). Ranges:

English	**0-800**
Metric	**0-2032**
Dots	**0-1535**

column — Distance from the reference *column* selected in the Graphics Header to the *bottom* of the box. Measure in units selected in the Graphics Header: 1/100 inches (English), 1/10 millimeters (Metric), or dots (Graphic). Ranges:

English	**0-400**
Metric	**0-1118**
Dots	**0-768**

algorithm — Coding method to use for bitmap data (see Chapter 2).

- **H** Hex Representation
- **R** Run Length Encoding

"data" — Character string derived under "Hex Representation" or "Run Length Encoding" procedures in Chapter 2, "Creating the Graphic Image." Do not put spaces or any other character between the hex numbers or run length code letters.

Example

```
B,46,1,H,"000000400000" ¦
```

- **B**itmap image field
- The image begins **46** dots from the bottom of the graphic area
- The image begins **1** dot from the left edge of the graphic area
- **H**ex representation is used
- The top row of dots was translated from binary "00000000/00000000/00000000/01000000/00000000/00000000" to hex **"000000400000"**

Next-Bitmap Field

A Next-Bitmap field continues a sequence of bitmap information. This field allows you to write the bitmap data without having to re-calculate row and column locations.

Syntax

```
N,adjdir,adjamt,algorithm,"data" ¦
```

N Token to indicate a next bitmap field.

adjdir Direction to adjust this row of data from the previous bitmap or next-bitmap field.

- **0** Adjust toward top of graphic image from the previous bitmap field.
- **1** Adjust toward bottom of graphic image from the previous bitmap field.

adjamt Amount of row adjustment, in dot rows (usually 1). You can elongate an image by adjusting by two or more rows. The elongated image will appear to have white horizontal lines running through it.

algorithm Coding method for bitmap data (see Chapter 2, "Creating the Graphic Image").

- **H** Hex Representation
- **R** Run Length Encoding

"data" Character string derived under "Hex Representation" or "Run Length Encoding" procedures in Chapter 2, "Creating the Graphic Image."

Example

```
N,1,1,H,"000000E00000"
```

- **N**ext Bitmap field
- The Bitmap field began at the top of the image The Next-Bitmap field adjusts toward the bottom of the supply (**1**)
- The amount of adjustment is **1** row
- **H**ex representation is used
- The second row of dots translates from binary "0000000/00000000/00000000/11100000/00000000/00000000" to hex **"000000E00000"**

Duplicate Field

If a line of data is identical to a Bitmap or Next-Bitmap field, the Duplicate field allows you to repeat the dot sequence without re-typing the data. It represents one row of dots on the image.

Syntax

```
D,adjdir,adjamt,count ¦
```

D — Token to indicate a Duplicate field.

adjdir — Direction to adjust this row of data from the previous bitmap or next-bitmap field.

- **0** Adjust toward top of supply from the previous bitmap field.
- **1** Adjust toward bottom of supply from the previous bitmap field.

adjamt — Amount of row adjustment, in dots (rows).

count — Number of times to duplicate the field. The maximum is 250.

Example

```
D,1,1,2 ¦
```

- Rows 3 and 4 are duplicates of row 2
- The image is being constructed from the top down, so adjustment is toward the bottom of the supply (**1**)
- Adjustment is made in **1** row increments
- The row is repeated **2** times.

Constant Text Fields

Constant text is a set of fixed characters that prints on all supplies using that format. The user cannot change the characters. Constant text can appear either in graphics or formats.

Syntax

```
C,row,column,gap,font,hgt.mag.,wid.mag.,
color,align, char-rot.,field-rot.,
"fixed char"
```

C1.*C* Indicates a constant text field.

C2.*row* Distance from bottom of print area to lower left corner of field. (Bottom exits the printer first.) Ranges:

English	**0-800**
Metric	**0-2032**
Dots	**0-1535**

C3.*column* Distance to lower left corner of field, from left edge of print area. Ranges:

English	**0-400**
Metric	**0-1118**
Dots	**0-768**

C4.*gap* Number of dots between characters. **0-99**.

C5.*font* Style of font.

1 Standard **3** Bold
2 Reduced

The bold font does not accept lower-case letters.

C6.*hgt. mag.* Height magnifier, **1-7** (times).

C7.*wid. mag.* Width magnifier, **1-7** (times).

C8.*color* Values:

B Black characters
W White characters on black background

C9.*alignment* Enter **L**. *Alignment* is constant.

C10.*char rot.* Character rotation. Values:

- **0** Top of character points to top of field
- **1** Top of character points to left of field
- **2** Top of character points to bottom of field
- **3** Top of character points to right of field

C11.*field rot.* Field rotation. Lower left corner of field is the pivot point.

- **0** Top of overlay points to top of supply
- **1** Top of overlay points to left of supply
- **2** Top of overlay points to bottom of supply
- **3** Top of overlay points to right of supply

C12.*fixed char.* Fixed characters to appear in the field. Maximum **200** characters. Surround with quotation marks.

Example

```
C,350,40,0,1,1,1,B,L,0,0,"Dry Clean Only"
```

- **C**onstant text
- Lower left corner of field at row = **3.5 inches**
- Column = **0.4 inches**
- Default spacing (**0**)
- Standard font (**1**)
- Height magnification **1**x
- Width magnification **1**x
- **B**lack characters
- Characters aligned **l**eft
- No character rotation (**0**)
- No field rotation (**0**)
- **"Dry Clean Only"** prints on all tags.

Line Fields

The line field in a graphic is identical to the line field in a standard format.

Syntax

```
L,S,row,column,end row,end column,thickness,pattern ¦
or
L,V,row,column,angle,length,thickness,pattern ¦
```

L — Token to indicate a line field.

type — Type of line. Only vertical and horizontal lines are supported. Values:

- **S** Segment. Choose the starting point and the ending point.
- **V** Vector. Choose the starting point, the angle, and the length.

row — Distance from the reference *row* selected in the Graphics Header to the line origin. Measure in units selected in the Graphics Header: 1/100 inches (English), 1/10 millimeters (Metric), or dots (Graphic). Ranges:

English	**0-800**
Metric	**0-2032**
Dots	**0-1535**

column — Distance from the reference *column* selected in the Graphics Header to the line *origin*. Measure in units selected in the Graphics Header: 1/100 inches (English), 1/10 millimeters (Metric), or dots (Graphic). Ranges:

English	**0-400**
Metric	**0-1118**
Dots	**0-768**

end row/angle

If using Segments
Distance from the reference *row* selected in the Graphics Header to the *end point* of the line. Ranges same as *row*. On horizontal lines, this value must match the *row* value.

If Using Vectors
Angle of line. Values: 0, 90, 180, or 270.

end column/ length

If using Segments
Distance from the reference *column* selected in the Graphics Header to the *end point* of the line. Ranges same as *column*. On vertical lines, this value must match the *column* value.

If using Vectors
Length of the line, in units selected in the Graphics Header. Ranges same as *row* or *column*.

thickness Line thickness 0-99, in dots.

pattern Line pattern. The 9406 supports only one pattern (solid line). Enter "" only.

Example

```
L,S,110,30,110,150,10,"" ¦
```

- **L**ine
- Line defined as a **S**egment
- Origin row **1.1** inches from reference row
- Origin column **0.3** inches from reference column
- End point **1.1** inches from reference row
- End point **1.5** inches from reference column
- **10** dots thick
- Solid line (**""**).

Box (Quadrilateral) Field

The box (quadrilateral) field in a graphic is identical to the quadrilateral field in a standard format.

Syntax

```
Q,row,column,endrow,endcol,thickness,pattern ¦
```

Q Token to indicate a box (quadrilateral) field.

row Distance from the reference *row* selected in the Graphics Header to the *lower left corner* of the box. Measure in units selected in the Graphics Header: 1/100 inches (English), 1/10 millimeters (Metric), or dots (Graphic). Ranges:

English	**0-800**
Metric	**0-2032**
Dots	**0-1535**

column Distance from the reference *column* selected in the Graphics Header to the *lower left corner* of the box. Measure in units selected in the Graphics Header: 1/100 inches (English), 1/10 millimeters (Metric), or dots (Graphic). Ranges:

English	**0-400**
Metric	**0-1118**
Dots	**0-768**

end row Distance from the reference *row* selected in the Graphics Header to the *upper edge* of the box. Ranges same as *row*.

end col Distance from the reference *column* selected in the Graphics Header to the *right edge* of the box. Ranges same as *column*.

thickness Line thickness 0-99, in dots. Thickness fills to the center of the box.

pattern Line pattern. Only one pattern (solid line) supported on 9406. Enter "" only.

Example

```
Q,240,30,270,150,3,""
```

- Box (**Q**uadrilateral)
- Lower left corner **2.4** inches from reference row
- Lower left corner **0.3** inches from reference column
- Upper right corner **2.7** inches from reference row
- Upper right corner **1.5** inches from reference column
- Line thickness **3** dots
- Solid line (**""**).

Example Bitmap

```
                                                    Row(s)
{G,5,A,T,G,96,48,0,"LETTER A"¦
B,46,1,H,"000006000000"¦                              1
N,1,1,H,"00000F000000"¦                               2
D,1,1,1¦                                              3
N,1,1,H,"00001F800000"¦                               4
D,1,1,1¦                                              5
N,1,1,H,"00003FC00000"¦                               6
D,1,1,2¦                                            7, 8
N,1,1,H,"00007FE00000"¦                               9
D,1,1,1¦                                             10
N,1,1,H,"0000FFF00000"¦                              11
D,1,1,2¦                                           12, 13
N,1,1,H,"0001FFF80000"¦                              14
N,1,1,H,"0001C1C00000"¦                              15
N,1,1,H,"0003C1FC0000"¦                              16
N,1,1,H,"0003F0FC0000"¦                              17
N,1,1,H,"0007F0FE0000"¦                              18
D,1,1,1¦                                             19
N,1,1,H,"000FF0FF0000"¦                              20
N,1,1,H,"000FE07F0000"¦                              21
D,1,1,1¦                                             22
N,1,1,H,"001FE07F8000"¦                              23
D,1,1,1¦                                             24
N,1,1,H,"003FC03FC000"¦                              25
D,1,1,2¦                                           26, 27
N,1,1,H,"007FFFFFE000"¦                              28
D,1,1,1¦                                             29
N,1,1,H,"00FFFFFFF000"¦                              30
D,1,1,1¦                                             31
N,1,1,H,"01FFFFFFF800"¦                              32
D,1,1,1¦                                             33
N,1,1,H,"03FFFFFFFC00"¦                              34
D,1,1,1¦                                             35
N,1,1,H,"03FC0003FC00"¦                              36
N,1,1,H,"07F80001FE00"¦                              37
D,1,1,1¦                                             38
N,1,1,H,"0FF80001FF00"¦                              39
N,1,1,H,"0FF00000FF00"¦                              40
D,1,1,1¦                                             41
N,1,1,H,"1FF00000FF80"¦                              42
N,1,1,H,"1FE000007F80"¦                              43
N,1,1,H,"3FE000007FC0"¦                              44
D,1,1,2¦                                           45, 46
}
```

4. Placing the Graphic in a Format

Perform these tasks to print a graphic image within a format.

1. Create the graphic image as shown in Chapter 3, "Defining Graphic Images."
2. Send the graphic to the printer.
3. If the graphic will be stored in RAM, place a field definition in the format to reference the graphic.
4. Send the check digit (if necessary), format, and batch to the printer.

The information placed in the format is called a *graphic field definition*.

The graphic field references the graphic image by the *GraphID* in the Graphic Header. This field definition is required only if the graphic will be stored in RAM.

Syntax

```
G,graphID,refrow,refcol,mode,rotation ¦
```

G	Header to indicate a graphic field definition.
graphID	The GraphID number (1 to 30) of the graphic image. Matches the GraphID in the Graphics Header.
refrow	The distance from the *bottom* of the *print area* on the supply to the *bottom* of the *graphic image*. Use the units of measure specified in the *Format* Header. This value overrides the reference row in the Graphics Header. Ranges: English **0-800** Metric **0-2032** Dots **0-1535**

NOTE: Add the row specified in the Constant Text, Bitmap, Line, or Box field definition to *refrow* to determine the actual position in the print image.

refcol The distance between the *left* edge of the *print area* on the supply and the *left* edge of the *graphic image*. Use the units of measure specified in the Format Header. This value overrides the reference column in the Graphics Header. Ranges:

English **0-400**

Metric **0-1118**

Dots **0-768**

NOTE: Add the column specified in the Constant Text, Bitmap, Line, or Box field definition to *refcol* to determine the actual position in the print image.

mode Imaging mode. Enter **0**.

rotation The orientation of the graphic on the supply. Enter **0**.

Example

```
G,5,075,025,0,0
```

- Place a **G**raphic image in this format.
- The ID number of the graphic image is **5**.
- Place the lower left corner of the graphic **0.75** inches from the bottom of the print area.
- Place the lower left corner of the graphic **0.25** inches from the left edge of the print area.
- Standard imaging mode is used (**0**).
- The graphic is placed in standard orientation on the supply (**0**).

Appendix A. Conversion Charts

Use the charts in this appendix to convert dot sequences from the image dot pattern to data stream codes.

Use the Binary to Hex Conversion Chart to convert binary dot sequences to hexadecimal numbers for bitmap data streams.

Use the Run Length Encoding Chart to convert dot sequences to alphabetic characters for bitmap data streams.

Binary to Hex Conversion Chart

Binary	Hex	Binary	Hex
00000000	00	00010000	10
00000001	01	00010001	11
00000010	02	00010010	12
00000011	03	00010011	13
00000100	04	00010100	14
00000101	05	00010101	15
00000110	06	00010110	16
00000111	07	00010111	17
00001000	08	00011000	18
00001001	09	00011001	19
00001010	0A	00011010	1A
00001011	0B	00011011	1B
00001100	0C	00011100	1C
00001101	0D	00011101	1D
00001110	0E	00011110	1E
00001111	0F	00011111	1F

Binary	Hex	Binary	Hex
00100000	20	00110000	30
00100001	21	00110001	31
00100010	22	00110010	32
00100011	23	00110011	33
00100100	24	00110100	34
00100101	25	00110101	35
00100110	26	00110110	36
00100111	27	00110111	37
00101000	28	00111000	38
00101001	29	00111001	39
00101010	2A	00111010	3A
00101011	2B	00111011	3B
00101100	2C	00111100	3C
00101101	2D	00111101	3D
00101110	2E	00111110	3E
00101111	2F	00111111	3F
01000000	40	01010000	50
01000001	41	01010001	51
01000010	42	01010010	52
01000011	43	01010011	53
01000100	44	01010100	54
01000101	45	01010101	55
01000110	46	01010110	56
01000111	47	01010111	57
01001000	48	01011000	58
01001001	49	01011001	59
01001010	4A	01011010	5A
01001011	4B	01011011	5B
01001100	4C	01011100	5C
01001101	4D	01011101	5D
01001110	4E	01011110	5E
01001111	4F	01011111	5F

Binary	Hex	Binary	Hex
01100000	60	01110000	70
01100001	61	01110001	71
01100010	62	01110010	72
01100011	63	01110011	73
01100100	64	01110100	74
01100101	65	01110101	75
01100110	66	01110110	76
01100111	67	01110111	77
01101000	68	01111000	78
01101001	69	01111001	79
01101010	6A	01111010	7A
01101011	6B	01111011	7B
01101100	6C	01111100	7C
01101101	6D	01111101	7D
01101110	6E	01111110	7E
01101111	6F	01111111	7F
10000000	80	10010000	90
10000001	81	10010001	91
10000010	82	10010010	92
10000011	83	10010011	93
10000100	84	10010100	94
10000101	85	10010101	95
10000110	86	10010110	96
10000111	87	10010111	97
10001000	88	10011000	98
10001001	89	10011001	99
10001010	8A	10011010	9A
10001011	8B	10011011	9B
10001100	8C	10011100	9C
10001101	8D	10011101	9D
10001110	8E	10011110	9E
10001111	8F	10011111	9F

Binary	Hex	Binary	Hex
10100000	A0	10110000	B0
10100001	A1	10110001	B1
10100010	A2	10110010	B2
10100011	A3	10110011	B3
10100100	A4	10110100	B4
10100101	A5	10110101	B5
10100110	A6	10110110	B6
10100111	A7	10110111	B7
10101000	A8	10111000	B8
10101001	A9	10111001	B9
10101010	AA	10111010	BA
10101011	AB	10111011	BB
10101100	AC	10111100	BC
10101101	AD	10111101	BD
10101110	AE	10111110	BE
10101111	AF	10111111	BF
11000000	C0	11010000	D0
11000001	C1	11010001	D1
11000010	C2	11010010	D2
11000011	C3	11010011	D3
11000100	C4	11010100	D4
11000101	C5	11010101	D5
11000110	C6	11010110	D6
11000111	C7	11010111	D7
11001000	C8	11011000	D8
11001001	C9	11011001	D9
11001010	CA	11011010	DA
11001011	CB	11011011	DB
11001100	CC	11011100	DC
11001101	CD	11011101	DD
11001110	CE	11011110	DE
11001111	CF	11011111	DF

Binary	Hex	Binary	Hex
11100000	E0	11110000	F0
11100001	E1	11110001	F1
11100010	E2	11110010	F2
11100011	E3	11110011	F3
11100100	E4	11110100	F4
11100101	E5	11110101	F5
11100110	E6	11110110	F6
11100111	E7	11110111	F7
11101000	E8	11111000	F8
11101001	E9	11111001	F9
11101010	EA	11111010	FA
11101011	EB	11111011	FB
11101100	EC	11111100	FC
11101101	ED	11111101	FD
11101110	EE	11111110	FE
11101111	EF	11111111	FF

Run Length Encoding Chart

Black Dots			
# of Dots	Code	# of Dots	Code
1	A	14	N
2	B	15	O
3	C	16	P
4	D	17	Q
5	E	18	R
6	F	19	S
7	G	20	T
8	H	21	U
9	I	22	V
10	J	23	W
11	K	24	X
12	L	25	Y
13	M	26	Z

White Dots			
# of Dots	Code	# of Dots	Code
1	a	14	n
2	b	15	o
3	c	16	p
4	d	17	q
5	e	18	r
6	f	19	s
7	g	20	t
8	h	21	u
9	i	22	v
10	j	23	w
11	k	24	x
12	l	25	y
13	m	26	z

For supplies, service, or assistance call:

TOLL FREE:
1-800-543-6650 (In the U.S.A.)
1-800-263-4650 (In Canada)

Printed by the Techical Communications Department
Printed in the U.S.A.